PLL implements FPGA-based SDRAM controller

EDDY DEBAERE, BARCO GRAPHICS, GHENT, BELGIUM

As FPGA capabilities increase and time to market decreases, FPGAs gain more acceptance for implementing both data and control paths. Thus, they find wide use as controllers and datapath glue logic for fast-page DRAMs. Synchronous DRAMs (SDRAMs), whose control signals use a clock input as reference, are a natural target for FPGA-based controllers. SDRAMs operate at frequencies of 100 MHz and higher (in contrast with fast-page DRAMs, for which a 60-MHz memory-system clock was considered high). **Figure 1** shows a way to implement FPGA-based SDRAM controllers. **Figure 2** shows the timing for a Xilinx XC4010E-2 device. You can apply the method to FPGAs from other vendors, as well as to high-frequency systems other than SDRAMs.

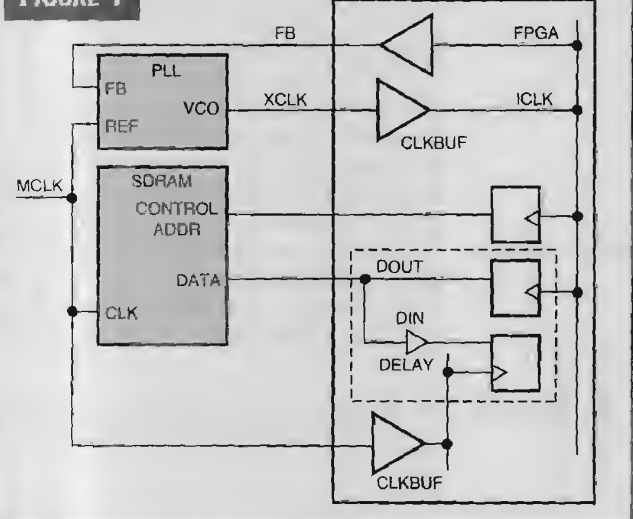
When you use FPGAs, delays to get on and off chip add up quickly: clock pin to internal clock buffer for the internal clock-distribution net (5.4 nsec) and internal clock to output flip-flop (4.5 nsec minimum). The sum of these delays excludes an FPGA from application with SDRAMs using a 10-nsec clock period, considering a 3-nsec setup time for the address, control, and data-out signals. **Figure 1** shows how to use a PLL with an FPGA and how to implement clock and control-path signals to make FPGA-based SDRAM controllers that operate at 100 MHz and beyond. The SDRAM address, data, and control signals use the memory-system clock, MCLK, as reference and the FB signal from an FPGA output pin as feedback.

Because the PLL loop uses no dividers, MCLK and XCLK have the same frequency. When the PLL locks, the XCLK signal precedes MCLK by a time equal to the sum of the clock-buffer delay (from XCLK to ICLK) and the combinatorial output-buffer delay (from ICLK to the PLL feedback pin). The address, control, and data-out signals obtain their clock signals from ICLK. Because the delay from ICLK to the clocked output pin (7.0 nsec for the slew-rate-limited output) is close to but higher than the delay from ICLK to the FB output pin (4.8 nsec for a fast output), the clocked output appears early in the CLK period.

The required SDRA setup time is thus easily fulfilled in a 100-MHz system: (10–(7.0–4.8))>3 nsec. Note that changes in temperature, IC processes, and voltage have little influence on performance, because the XCLK clock-generating circuit is a closed-loop system. Moreover, the output buffer of the FB pin and the control and data pins have the same clock input as does the input pin, and all circuit blocks reside on the same die. Tests designed to check the data-input setup-and-hold times used a clocked input flip-flop (in the same I/O block as the data-output flip-flop), which received its clock from an internal version of MCLK. This configuration avoids excessive hold-time requirements for the SDRAM. *(Editor's note: An* EDN *contributing editor warns that the 7.0–4.8 nsec reflects* maximum *specified times for the XC4010E-2 and yields an acceptable margin for an SDRAM with 1.5-nsec minimum hold time. However, an FPGA running at* typical *specs may present a marginal situation.)* Tests using the AV9170-01 PLL from ICS show satisfactory performance to 106.25 MHz. (DI#2165) EDN

To Vote For This Design, Circle No. 415

FIGURE 1



The use of a PLL with an FPGA eliminates the setup-time problems inherent in an FPGA-only configuration, allowing you to design SDRAM controllers for clock frequencies of 100 MHz and beyond.

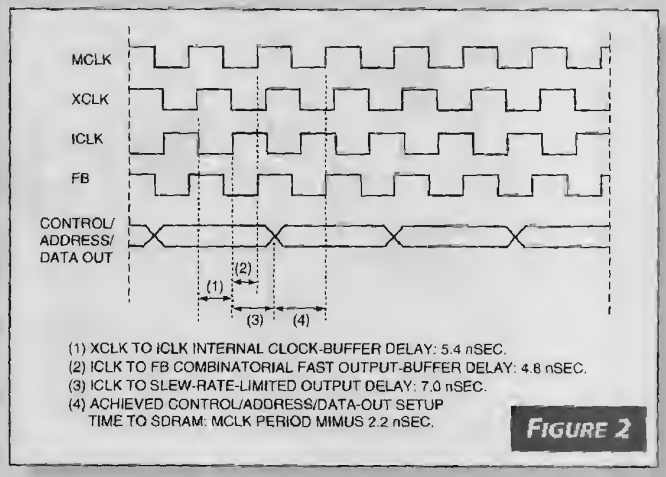


FIGURE 2

The timing for a Xilinx XC4010E-2 device shows 7.0–4.8=2.2 nsec, an acceptable figure for an SDRAM with 1.5-nsec minimum hold time.

Comparator uses signal-dependent hysteresis

P KREHLIK AND L SLIWCZYNSKI, UNIVERSITY OF MINING AND METALLURGY, KRAKOW, POLAND

Sometimes, you need to distinguish between two voltages, using some hysteresis in the decision. When the levels of the compared signals vary over a wide range (for example, a few orders of magnitude), the hysteresis width should vary similarly to ensure a constant ratio between hysteresis width and signal level. You could encounter such a situation, for example, when you need to decide which of two transmission channels conveys a "higher quality" signal (one with a higher level). You need the hysteresis to avoid permanent changes in the decision when the signal levels are close to each other. In this case, the best way to choose the greater voltage is to make the decision by taking the ratio of the signals instead of comparing them directly. **Figure 1** shows a circuit with signal-variant hysteresis.

Amplifiers IC_{1A} and IC_{1B} with associated resistors and diodes, R_1, R_2, D_1, and D_2, operate as logarithmic converters, producing output voltages

$$V_{O1,O2} = -V_T \ln\left[\frac{V_{I1,I2}}{RI_S}\right],$$

where $V_T = kT/q \sim 25$ mV at room temperature, and I_S is the saturation current of the p-n junction. The derivation of the equation assumes that $R_1 = R_2 = R$. Next, voltages V_{O1} and V_{O2} combine in the summing amplifier, IC_{1C}. Using the assumption that $R_3 = R_5$ and $R_4 = R_6$ and that the diodes are matched, the output voltage from IC_{1C} is

$$V_{O3} = \frac{R_4}{R_3}(V_{O2} - V_{O1}) = V_T \frac{R_4}{R_3} \ln\left[\frac{V_{I1}}{V_{I2}}\right].$$

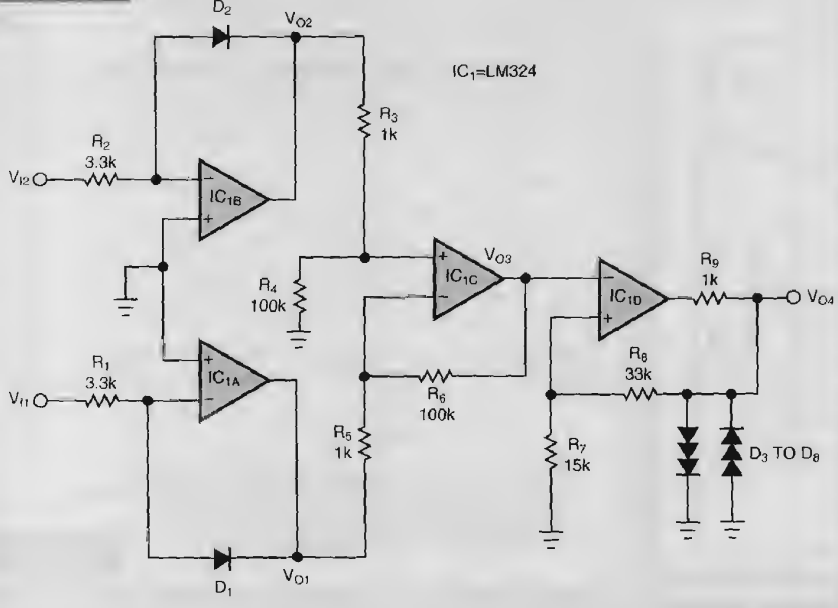
The voltage is thus proportional to the ratio of the input voltages. The last amplifier in the circuit, IC_{1D}, is a standard inverting comparator with hysteresis centered around zero and the threshold levels:

$$V_H^+ = V_{O4}^+ \frac{R_7}{R_7 + R_8}, \text{ and } V_H^- = V_{O4}^- \frac{R_7}{R_7 + R_8}.$$

The voltage swing at the output of the circuit (V^+_{O4}, V_{-O4}), is a function of the limiter comprising resistor R_9 and diodes D_3 to D_8. The output voltage, V_{O4}, changes state when $V_{O3} > V^+_H$ (output goes low) or $V_{O3} < V^-_H$ (output goes high). These conditions correspond to

$$V_{I1} > V_{I2} \exp\left[\frac{V_H^+}{V_T} \bullet \frac{R_3}{R_4}\right], \text{ or } V_{I1} < V_{I2} \exp\left[\frac{V_H^-}{V_T} \bullet \frac{R_3}{R_4}\right].$$

FIGURE 1



Signal-variant hysteresis allows you to compare signals with a constant-ratio trip window for a wide range of signal levels.

The design allows you to distinguish between two signals, with a tolerance set by the external components, R_3, R_4, R_7, and R_8. When $V^+_{O4} = |V^-_{O4}|$, the tolerance is symmetrical. With the given values, the output state changes when one input signal is approximately 30% higher than the other. If you need to use the circuit over a wide temperature range, you should take the thermal dependence of the threshold levels into consideration. The circuit uses an LM324 quad op amp. If you use a rail-to-rail amplifier (for example, the LMC6484), the output limiter is unnecessary. (DI #2166) EDN

To Vote For This Design, Circle No. 416

Simple LCD interface takes two wires

ED MASTE, JEM DESIGNS, PICKERING, ON, CANADA

Alphanumeric LCD modules can provide an attractive display for a project, but their parallel I/O lines require a large number of outputs from a µC. For example, a direct LCD interface would consume all of the I/O pins of a small µC, such as the eight-pin PIC12C508 from Microchip Technology (Chandler, AZ).

However, the circuit in **Figure 1a** implements a clocked serial input for an LCD module and allows a µC to communicate with the LCD over just two signal lines. You can fit the circuit onto a small pc board and mount the board directly behind the LCD module. The connection between the board and LCD module comprises just four wires, including V_{CC} and ground.

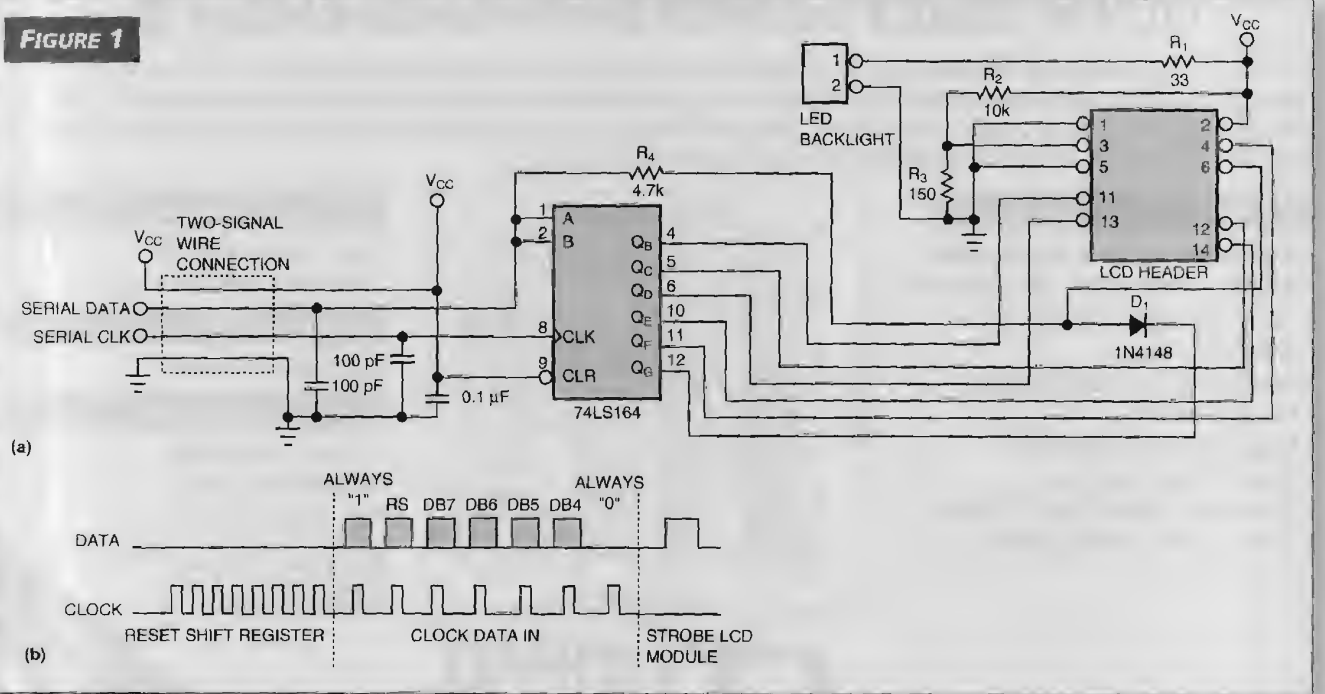
A 74LS164 serial-to-parallel shift register forms the heart of the circuit and communicates with the LCD module in 4-bit mode. The shift register's outputs directly drive the LCD module's data inputs, DB7 through DB4, as well as RS, the control/data select input. The only signal that is not a direct output from the shift register is the LCD module's enable pin, which is Pin 6. R_4 and D_1 derive the enable input and ensure that the enable signal remains low until valid data is present at all other outputs. R_1 is a current-limiting resistor for an LED backlight if the LCD module has one, and R_2 and R_3 set the contrast level of the module. You can replace R_2 and R_3 with a potentiometer if an adjustable contrast level is desirable.

The µC must first clock in at least seven zeros for the first write (**Figure 1b**). This series of zeros guarantees that output Q_G of IC_1 is low and will remain so for the next six clock cycles. This low level ensures that the enable input (Pin 6 of the LCD module) remains low because of diode D_1. The serial-input stream then must consist of a one, followed by RS, and finally DB_7 through DB_4. With the data input low, an additional clock pulse shifts in a zero and lines up all of the outputs to the LCD module with the correct shift-register pins. The first clocked-in one now appears at Q_G, which reverse-biases D_1. The enable continues to remain low because the data input to the circuit is low, and this situation pulls the enable low through R_4. The data input then must go high for at least 450 nsec without a clock pulse to provide the enable pulse for the LCD module. This process, including clocking in at least six zeros, is then repeated for the next 4-bit write.

For further information, including minimum cycle times and 4-bit-mode communication, consult the technical documentation provided by an LCD module manufacturer, such as Optrex (Plymouth, MI). (DI #2177)

To Vote For This Design, Circle No. 417

FIGURE 1



Using a 74LS164 shift register and 4-bit communication, a µC can control an LCD module using just two lines: serial data and clock (a). The µC must first clock a series of zeros to the circuit, followed by a one, RS, and DB_7 through DB_4 (b).

Inexpensive logic controls stepper motor

DAVID ELLIS, ELLIS LINDAUER, PULLMAN, WA

A number of sophisticated ICs for stepper-motor control are now available. However, the advanced features of these chips—self-clocking, high-current drive, and full-step, half-step, and direction control—are often unnecessary or remain unused. For a design that needs to control only the number of steps, drive speed, and direction, you can make a very simple and inexpensive driver using two low-level logic chips (**Figure 1**). The cost of this controller is less than $1; the cost of dedicated motor-control ICs starts at around $5. The drawback is a slight increase in board space.

TABLE 1—STEPPER-MOTOR-DRIVE SEQUENCE

(a)

	State	A	B	C	D
CCW ↑	1	ON	OFF	ON	OF
	2	ON	OFF	OFF	ON
	3	OFF	ON	OFF	ON
	4	OFF	ON	ON	OFF

(b)

	State	A	B	C	D
↓ CW	1	1	0	1	0
	2	1	0	0	1
	3	0	1	0	1
	4	0	1	1	0

Going back to the basics, you can control a standard stepper-motor drive, whether bipolar or unipolar, using a four-step sequence (**Table 1a**). By replacing the on and off states with ones and zeros, respectively (**Table 1b**), Column B becomes the logical inverse of Column A, and Column D becomes the logical inverse of Column C. Thus, the corresponding state diagram (**Figure 2**) comprises just 2 bits. Clockwise rotation results from using a logical one to move sequentially from state one to state four and back to state one. Likewise, counterclockwise rotation results from using a logical zero to move through the states in the reverse order.

You can then produce the present-state/next-state assignment (**Table 2**) and the next-state maps (**Figure 3**). Then, by inspection, the logical choice is to loop the state maps out for D flip-flops, which produces the following two logic equations:

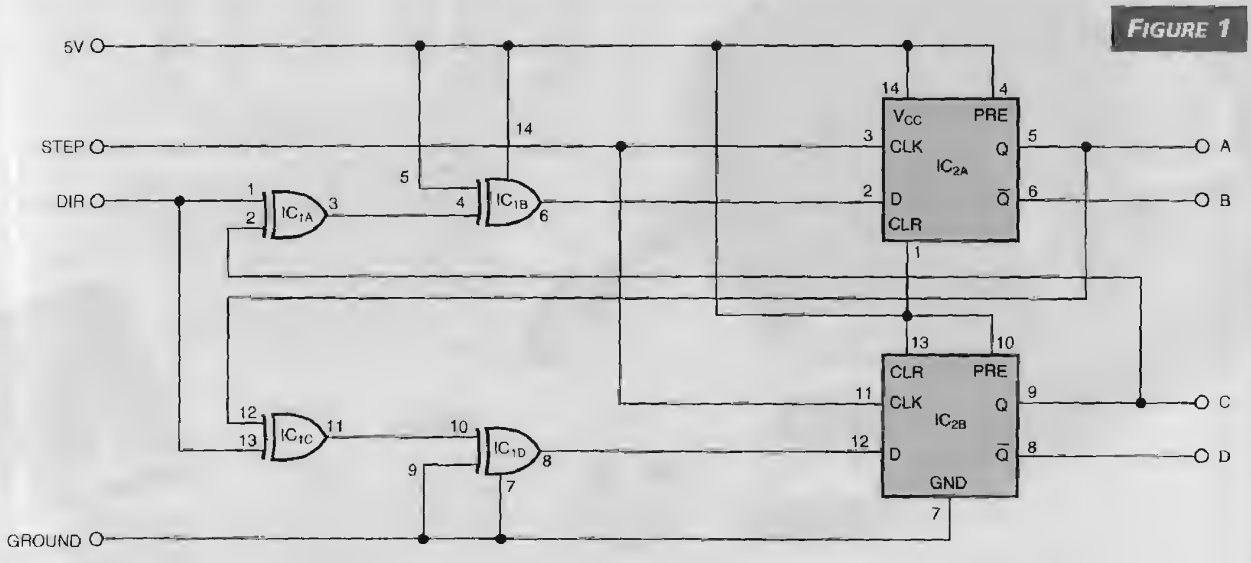
$$DA=((DIR)(C))+((DIR)(C)); \quad \textbf{(1)}$$

$$DB=((DIR)(A))+((DIR)(A)). \quad \textbf{(2)}$$

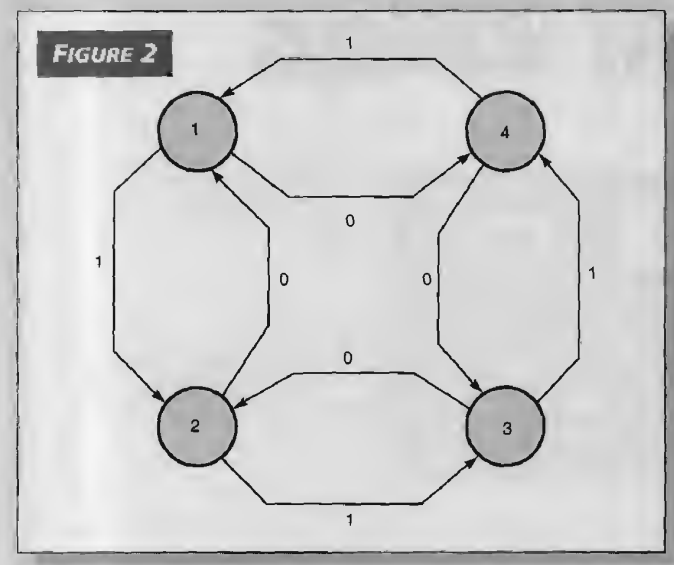
Equation 1 is an exclusive NOR, and **Equation 2** is an exclusive OR. To save space, you can use a single quad XOR chip to implement both equations. A dual D flip-flop completes the logic driver, as **Figure 1** shows. Using rising-edge-triggered D flip-flops helps keep the design simple while eliminating mode-change faults.

The circuit derives the four outputs from the Q and Q

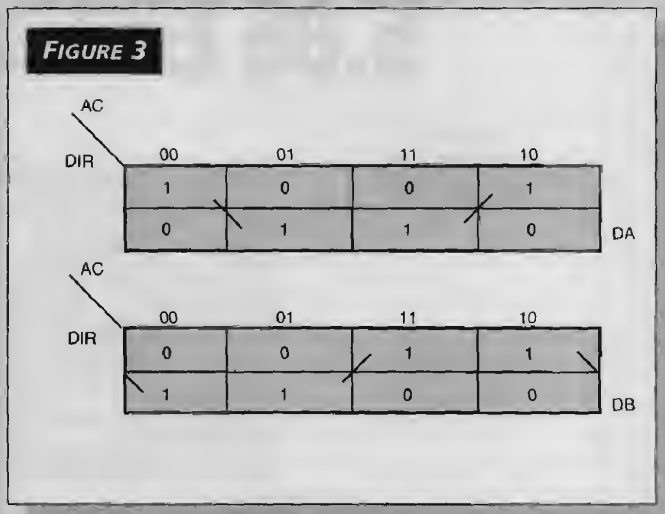
FIGURE 1



Two logic-level ICs can implement simple and inexpensive control of a stepper-motor driver.



The state diagram for the stepper-motor controller comprises just 2 bits.



The next-state maps correspond to two simple logic equations.

outputs of D flip-flops IC_{2A} and IC_{2B} in Figure 1. IC_{1A} XORs the Q output of flip-flop IC_{2B} with the DIR input, and the circuit transforms the output into an XNOR by using IC_{1B} as a controlled inverter. IC_{1B} then drives the D input of flip-flop IC_{2A}. Similarly, IC_{1C} XORs the Q output of IC_{2A} with the DIR input. The output of IC_{1C} drives XOR IC_{1D}, which acts as a noninverting buffer. The output of IC_{1D} drives the D input of IC_{2B}. Using XOR gate IC_{1D} as a buffer keeps the propagation delays to the D inputs of the flip-flops equal, which helps the circuit avoid any race conditions. The STEP signal is the step-rate input, which drives the clock inputs of both flip-flops.

The last design task is to add the appropriate-sized transistors to drive the stepper motor. In the case of the unipolar motor, output signals A, B, C, and D can directly drive the transistors. To drive a bipolar motor, you can use the A and C outputs to drive one-half of two H-bridges and the B and D outputs to drive the other corresponding half of the H-bridges. This design is possible because the B output is the inverse of A, and D is the inverse of C. (DI #2176)

EDN

To Vote For This Design, Circle No. 418

TABLE 2—PRESENT-STATE/NEXT-STATE ASSIGNMENT TABLE

Present state			Next state	
A	C	DIR	A	C
0	0	0	1	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

Low-voltage reset operates below 2.7V

BOB KELLY, MAXIM INTEGRATED PRODUCTS, SUNNYVALE, CA

New personal digital assistants, pagers, and other battery-powered systems operate at or below 2.7V, but power-on resets with thresholds below 2.6V are not commonly available. You can resolve this problem using a circuit that combines a 1.2V reference and a micropower regulator (**Figure 1a**). IC_1 integrates these two functions in a tiny SOT-143 package. A power-on-reset function must become active before the supply voltage reaches its nominal value, and IC_1 operates properly for supply voltages above 1.21V.

The R_1/R_2 divider and internal 1.204V reference establish a threshold that determines when the circuit asserts an active-low at the output. For the values in the **figure**, this threshold is 2.25V (**Figure 1b**). IC_1 has an open-drain output, so R_3 and C_1 control the length of the active-low pulse, RESET. In this case, the pulse length, or reset interval, is approximately 54 msec, which is sufficient reset time for

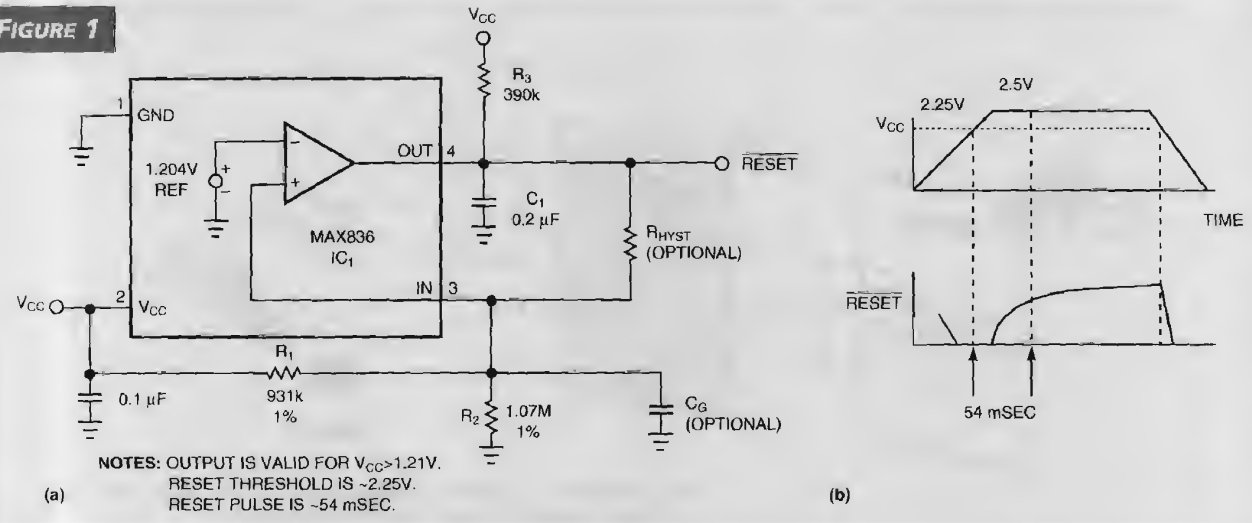
most µCs and other digital circuits.

Low power consumption distinguishes this circuit. The IC typically draws only 5 µA, and the R_1/R_2 divider draws slightly more than 1 µA in a 2.7V application. Pullup resistor R_3 consumes power only when the supply voltage droops out of tolerance, so the power loss is minimal in normal operation.

To prevent erratic behavior, IC_1 offers approximately 6 mV of built-in hysteresis. For more hysteresis, you can add a large-value resistor, R_{HYST}, between the IC's input and output; to reject short transients, IC_1 has an inherent glitch immunity of 35 µsec with 100 mV of overdrive. The input capacitance works with R_1 and R_2 to provide some lowpass-filter action. For further immunity from transients, which is unnecessary unless the power bus is noisy, you can form an additional lowpass filter by adding a small-value capacitor, C_G, to the input pin. (DI #2174)

To Vote For This Design, Circle No. 419

FIGURE 1



A 1.2V reference and micropower regulator in IC_1 (a) provide an active-low reset pulse of approximately 54 msec at power-up or when V_{CC} dips below 2.25V (b).

Alternating LED blinker uses four parts

ANDY MENG, CINCINNATI, OH

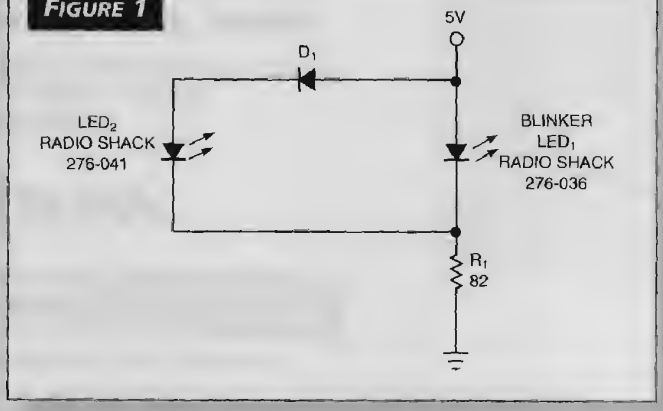
The circuit in **Figure 1** is an easy-to-make attention-getter and runs for a week or longer on two AA cells. In September, I used this circuit for a school fundraiser, and it helped me generate more than $100. My dad showed me a circuit in *EDN* that did the same thing, but it uses more parts (see "Alternating LED flasher uses minimal parts," *EDN*, Nov 20, 1997, pg 104).

The main element of this circuit is LED_1, a Radio Shack 276-036 blinking red LED. D_1 can be almost any silicon diode. The forward bias of D_1 brings the turn-on voltage of LED_2 up to 2.5V. R_1 is a current-limit resistor for LED_1, and this resistor also reduces the current of LED_1 for longer battery life.

LED_2 is a Radio Shack 276-041 red LED. When you apply power, LED_1 turns on and drops the voltage across LED_2 to 1V. When LED_1 turns off, the voltage across LED_2 equals 3V, and LED_2 turns on. (DI #2172)

To Vote For This Design, Circle No. 420

FIGURE 1



This blinking-LED circuit, designed by a seventh grader, uses only four parts.

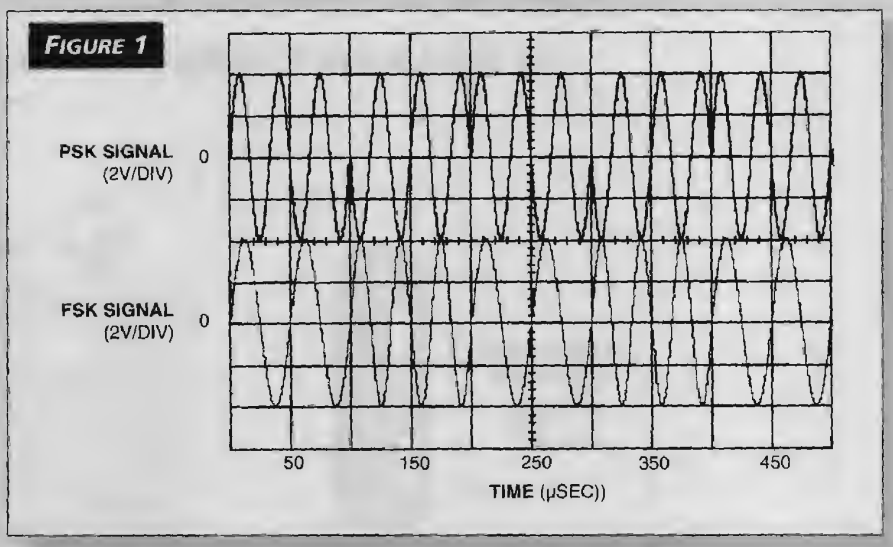
Spice generates PSK and FSK signals

DEBRA HORVITZ, GALAHAD SYSTEMS, LAGUNA HILLS, CA

Creating generators for amplitude-, frequency-, and phase-modulated signals can greatly simplify communication-system simulation. Although Spice includes a basic set of waveform generators, it includes no built-in support for many types of signals. You must create these signals from combinations of elements, and you can create variations of these built-in generators using Spice 2-dependent sources. However, using dependent sources to generate complex waveforms can require fairly complex Spice subcircuits.

Fortunately, the nonlinear, arbitrary dependent source—the B element—in Berkeley Spice 3 and IsSpice4 (Intusoft, San Pedro, CA) provides a quantum leap in capability over Spice 2-dependent sources. The B element is more versatile and easier to use. For example, **Listing 1** is the IsSpice4 subcircuit for a parameterized phase-shift-keying (PSK) source; **Listing 2** is the subcircuit of a parameterized frequency-shift-keying (FSK) source. **Figure 1** shows the resultant output signals of each subcircuit generator.

FIGURE 1



The behavioral and mathematical capabilities of Spice 3 and IsSpice 4 make it easy to create PSK and FSK signals.

The PSK subcircuit produces a coherent binary PSK signal according to the following equations:

$$s_1(t) = \sqrt{\frac{2 \cdot EB}{TB}} \cdot \sin(2\pi ft),$$

$$s_2(t) = -\sqrt{\frac{2 \cdot EB}{TB}} \cdot \sin(2\pi ft),$$

where EB is the transmitted energy per bit, TB is the bit duration, and f is the transmission frequency equal to NC/TB. (NC is an integer constant, and the period is 2×TB; thus, the duty cycle equals 50%.) In **Listing 1**, the input voltage source, VSIG, produces the polar form of the input signal. Using the B1 element, the subcircuit multiplies this input by the local oscillator voltage, VLO1, to produce the PSK output signal.

The subcircuit, FSK, produces a coherent binary FSK signal according to the following equations:

$$s_1(t) = \sqrt{\frac{2 \cdot EB}{TB}} \cdot \sin(2\pi f_1 t),$$

$$s_2(t) = \sqrt{\frac{2 \cdot EB}{TB}} \cdot \sin(2\pi f_2 t).$$

where f_1 is the high-bit transmission frequency, and f_2 is the low-bit transmission frequency. The frequency, f_1, is equal to NC+1/TB. The frequency, f_2, is equal to NC+2/TB. For simplicity, two pulse generators, VSIG and VSIGN, produce the input signal, m(t), and the inverse of the message signal, M(t), respectively. Again, using the B1 element, the subcircuit code multiplies these signals by the appropriate frequency generator: VLO1 for logic high and VLO2 for logic low. The sum of the resultant signals produces the FSK output signal. (DI #2173) EDN

To Vote For This Design, Circle No. 421

LISTING 1—ISSPICE4 SUBCIRCUIT FOR PSK SOURCE

```
.SUBCKT PSK  3   ; Signal Generator for PSK
VSIG 2 0  PULSE  {-(EB)^2}  {(EB)^2}  {TD}  {TR}  {TF}
+ {50/100*(2*TB-TR-TF)}  {2*TB}
B1 3 0 V = V(1) * V(2)
VLO1 1 0  SIN 0 {(2/EB)^2}  {(NC)/TB}
.ENDS
```

LISTING 2—SUBCIRCUIT FOR FSK SOURCE

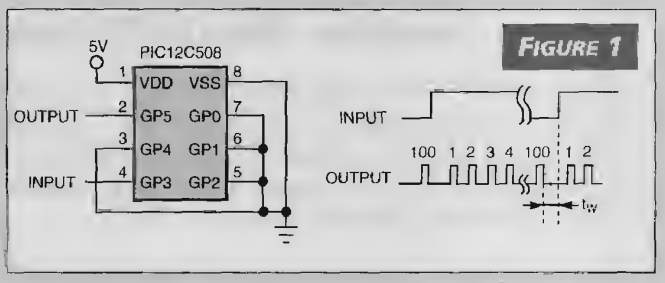
```
.SUBCKT FSK  5 ; Signal Generator for FSK
VSIG 3 0  PULSE  0  {(EB)^2}  {TD}  {TR}  {TF}
 {50/100*(2*TB-TR-TF)}  {2*TB}
VSIGN 4 0  PULSE  {(EB)^2}  0  {TD}  {TR}  {TF}
 {50/100*(2*TB-TR-TF)}  {2*TB}
B1 5 0 V = (V(1) * V(3)) + (V(2) * V(4))
VLO1 1 0  SIN  0  {(2/EB)^2}  {(NC+1)/TB}
VLO2 2 0  SIN  0  {(2/EB)^2}  {(NC+2)/TB}
.ENDS
```

Frequency multiplier improves line readings

YONGPING XIA, TELDATA INC, LOS ANGELES, CA

Because of the low frequencies involved, accurately measuring line-frequency variations is complicated. When you use an ordinary frequency counter with a 1-sec gate time, the reading would be 59, 60, or 61 Hz. To obtain 0.01-Hz accuracy, you must increase the gate time to 100 sec, a scale that most frequency counters do not offer. Moreover, reading updates are slow with this scale choice. One way to improve reading accuracy without sacrificing update time is to use a frequency multiplier. For instance, a frequency counter with a 1-sec gate time could provide 0.01-Hz accuracy if you multiply the line frequency by 100. Traditionally, you would configure the multiplier by using a VCO, a PLL, a frequency divider, and a lowpass filter. **Figure 1** shows another method, using only one component, to multiply the line frequency.

The PIC12C508 is an eight-pin, low-cost μC. It has a built-in 4-MHz oscillator and a reset circuit; it thus needs no external components. One of the μC's unique features is its software-based clock-frequency calibration. The controller dedicates one register, OSCCAL, to the calibration function. The upper 4 bits of OSCCAL accommodate 16 steps of calibration. Each step changes the clock frequency approximately 1.6%. The maximum change is approximately 25%. The line signal routes to the μC's GP3 pin. When the controller detects a low-to-high change in GP3, the program generates 100 pulses on GP5 and ignores the condition of GP3 (**Listing 1**). It then rechecks the status of GP3. A high value for GP3 means that the pulse generation is too slow; OSCCAL then increases by one step. If GP3 is low, then a counter operates until GP3 goes high.



An inexpensive μC, using no external components, allows you to measure line frequency with 0.01-Hz accuracy.

The number in the counter represents the delay (t_W) between the last pulse and the next line-signal change. The μC compares t_W with a fixed number. If t_W is larger than that number, indicating that the clock frequency is too high, the value in OSCCAL decreases by one step to slow down. If t_W is smaller than the fixed number, the value in OSCCAL increases by one step to speed up. Thus, the automatic clock-frequency adjustment continuously sends a signal at 100 times the line frequency. Note that, because of the calibration range, this approach has a limited input-frequency range. However, it accommodates line-frequency measurements from 58 to 62 Hz. You can download the assembly code from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2182. (DI #2182)

To Vote For This Design, Circle No. 422

LISTING 1—PIC12C508 CODE FOR LINE-FREQUENCY MEASUREMENT

```
        LIST    P=12C508

status          equ     0x03            ;
osccal          equ     0x05            ;
gpio            equ     0x06            ;
                                        ;
C               equ     0               ;
Z               equ     2               ;
                                        ;
cnt_1           equ     0x07            ;
cnt_2           equ     0x08            ;
cnt_3           equ     0x09            ;
cnt_4           equ     0x0a            ;
temp            equ     0x0b            ;
                                        ;
                org     0x0             ;
                                        ;
                movlw   0x80            ;
                movwf   osccal          ;
                movlw   0x1f            ;
                tris    gpio            ;
lp_1            btfsc   gpio, 3         ;       wait input low
                goto    lp_1            ;
lp_2            btfss   gpio, 3         ;       wait input high
                goto    lp_2            ;
lp_3            movlw   0x64            ;
                movwf   cnt_1           ;
lp_4                                    ;
                bsf     gpio, 5         ;       output high
                call    dly             ;
                nop                     ;
                nop                     ;
                bcf     gpio, 5         ;       output low
                decfsz  cnt_1, 1        ;
                goto    next_1          ;
                goto    last_cycle      ;
next_1          call    dly             ;
                goto    lp_4            ;
                                        ;
last_cycle                              ;
                clrf    cnt_3           ;
                clrf    cnt_4           ;
lp_5            incfsz  cnt_3, 1        ;
                goto    next_2          ;
                incf    cnt_4, 1        ;
next_2          btfss   gpio, 3         ;       wait input high
                goto    lp_5            ;
                movf    cnt_4, 0        ;
                btfss   status, Z       ;
                goto    freq_down       ;
                movlw   0x30            ;
                subwf   cnt_3, 0        ;
                btfss   status, C       ;
                goto    freq_up         ;
                goto    freq_down       ;
                                        ;
frsq_down                               ;       decrease frequency by 1.6%
                movlw   0xf0            ;
                andwf   osccal, 1       ;
                btfsc   status, Z       ;
                goto    lp_3            ;
                movlw   0x10            ;
                subwf   osccal, 1       ;
                goto    lp_3            ;
                                        ;
freq_up                                 ;       incrsase frequency by 1.6%
                movlw   0xf0            ;
                andwf   osccal, 1       ;
                movlw   0x10            ;
                addwf   osccal, 1       ;
                btfss   status, Z       ;
                goto    lp_3            ;
                subwf   osccal, 1       ;
                goto    lp_3            ;
                                        ;
dly                                     ;       delay loop
                movlw   0x18            ;
                movwf   cnt_2           ;
dly_1p          decfsz  cnt_2, 1        ;
                goto    dly_1p          ;
                retlw   0x00            ;
                                        ;
                end                     ;
```

In program mode (with S_1 grounded), the EA input is at logic "1." The PAL routes the read and write signals from the µC to the programmable device's RD and WR inputs, respectively. The µC executes instructions from its internal ROM. The ROM contains bootload program code that receives data from the RS-232C interface and programs the programmable memory device. Listing 1 shows the PAL equations. (DI #1834) EDN

To Vote For This Design, Circle No. 362

Supervisory IC makes simple line monitor

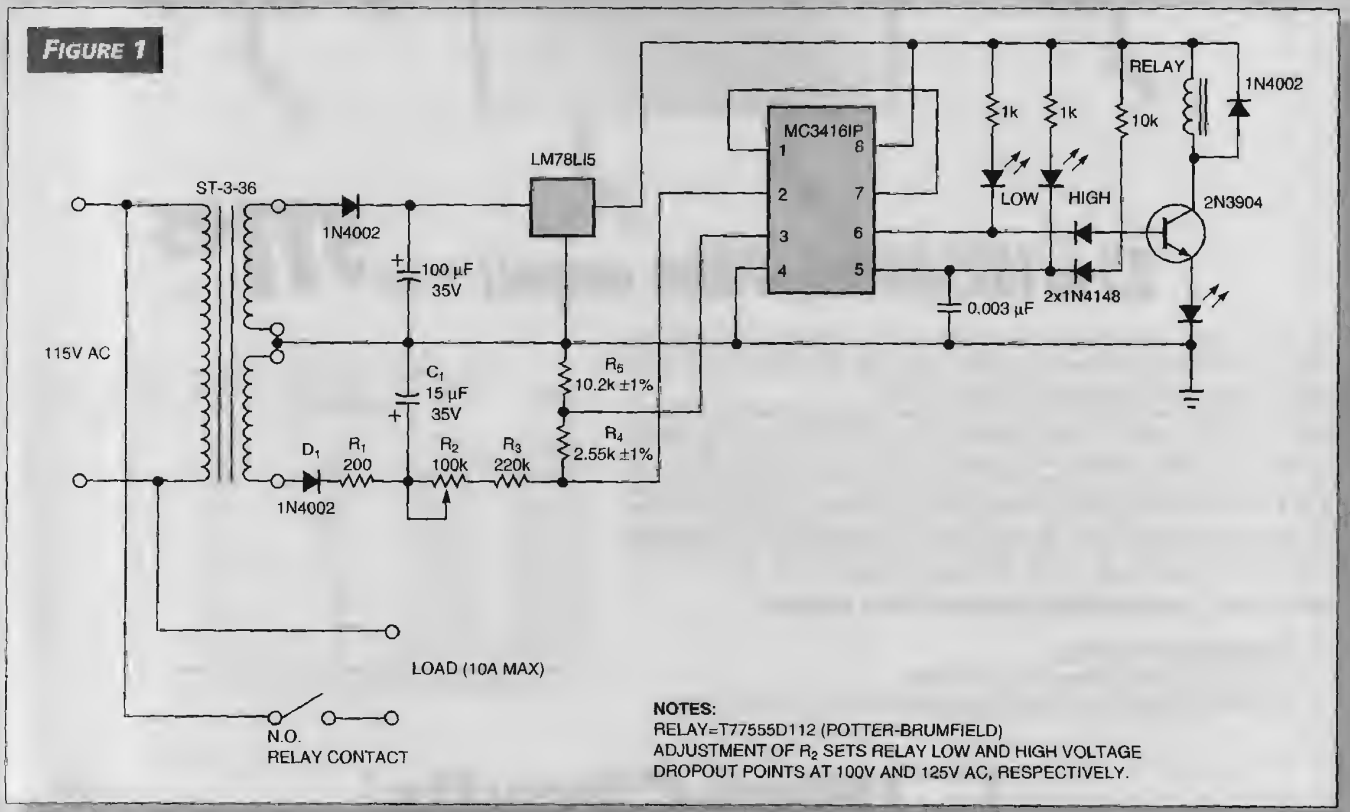
HENNO NORMET, TAVARES, FL

You can use the MC34161P universal voltage monitor to build a simple ac power-line monitor (a voltage-sensing relay). In this application (**Figure 1**), the mode select pin (pin 7) connects to the internal 2.55V reference (pin 1), and the IC functions as a positive-voltage window detector.

A small signal transformer provides total isolation from the power lines and performs a unique role in the circuit. During the positive half cycles of the line voltage, the transformer functions as a conventional power transformer feeding a 15V power supply through a half-wave rectifier. When the line voltage is negative, the transformer becomes a potential transformer accurately sensing the amplitude of the power line's voltage. During negative half cycles the transformer is essentially unloaded, thus eliminating all errors due to changes in loading (such as relay energizing or de-energizing).

The voltage-sensing circuit consists of R_1 through R_5, C_1, and D_1. The circuit responds to the peak value of the (assumed) sinusoidal line voltage. The response times to increasing and decreasing voltages are different; C_1 charges very quickly to the peak of the transformer's secondary voltage but discharges slowly through the voltage-divider network. This feature enables the voltage monitor to ignore momentary dips in line voltage and to quickly disconnect the load in case of overvoltage. Three LEDs provide visual indications of low, normal (in-band), or high line voltage. The relay de-energizes when the voltage is either too low or too high, which in turn disconnects the load from the line. (DI #1831) EDN

To Vote For This Design, Circle No. 363



This circuit uses the MC34161P universal voltage monitor as a simple power-line monitor.